

3 Data Warehouse

Os conceitos abordados nesse capítulo são importantes para facilitar o entendimento das abordagens apresentadas e analisadas nos próximos capítulos e permitir a compreensão do posicionamento do tema na literatura. Assim, o capítulo se inicia com a definição de *Data Warehouse* e dos benefícios trazidos por essa tecnologia. Posteriormente, define-se o processo de *Data Warehousing* e o problema da elicitação de requisitos para sistemas desse tipo é abordado.

Por fim, apresenta-se em linhas gerais a proposta de aprimoramento da tarefa de identificação dos requisitos de sistemas *data warehouses*, usando heurísticas voltadas para identificação de indicadores de desempenho e associação a processos de negócio.

3.1 Sistemas *Data Warehouse*

Os sistemas gerenciadores de banco de dados relacionais passam a ser amplamente produzidos e empregados nas organizações nos anos 80, demonstrando o grande potencial desses sistemas como repositórios de informações corporativas. Inmon (1997) foi um dos autores pioneiros no estabelecimento de uma definição formal para o termo *data warehouse*. Na sua concepção, esse termo corresponde a uma coleção de dados orientada por assunto, integrada, variante e não volátil, que tem por objetivo dar suporte aos processos de tomada de decisão.

O conceito de DW está diretamente relacionado à idéia de uma base de dados com diversos níveis de granularidade. Essa base é obtida a partir dos sistemas transacionais que servem de fonte de dados e tem como objetivo realizar o armazenamento dos mesmos em diferentes graus de relacionamento, facilitando e agilizando os processos de tomada de decisão pelos diversos níveis de gerência (Barbieri, 2001).

Segundo Paim (2003), os *data warehouses* são importantes por possibilitarem a transformação do dado operacional e heterogeneamente distribuído pela

organização em informação estratégica, suportando os processos de tomada de decisão e garantindo a competitividade requerida pelo negócio. Assim, os sistemas *data warehouses* (DW) respondem a questões que viabilizam a implementação de estratégias consideradas complexas sem esse apoio analítico. Além disso, o DW permite, dentre outros benefícios, a formulação, análise e controle da estratégia corporativa (Penna, 2004).

3.2 Data Warehousing

A obtenção de Data Warehouses baseia-se num processo de construção sistemático que abrange ferramentas, técnicas, algoritmos e uma arquitetura especialmente concebida para permitir o armazenamento de grandes volumes de dados, visando à obtenção de informações estratégicas. Esse processo de construção denomina-se *Data Warehousing* (DWing). Os sistemas gerados por esse processo são chamados de Sistemas de Suporte à Decisão (SSD) ou Sistemas *Data Warehouse* (Penna, 2004).

O processo de construção de DW, segundo Kimball *et. al* (1998) compreende um ciclo de vida que se inicia com o planejamento do projeto e definição dos requisitos de negócio. Essa tarefa é destacada aqui pela sua importância e relação com o contexto dessa pesquisa, já que a descoberta das necessidades de informação analíticas dos tomadores de decisão, bem como a disponibilidade e qualidade da informação proveniente dos sistemas transacionais, tornam-se fundamentais para sucessos em iniciativas de DWing. As demais tarefas desse ciclo de vida estão ilustradas abaixo na Figura 7.

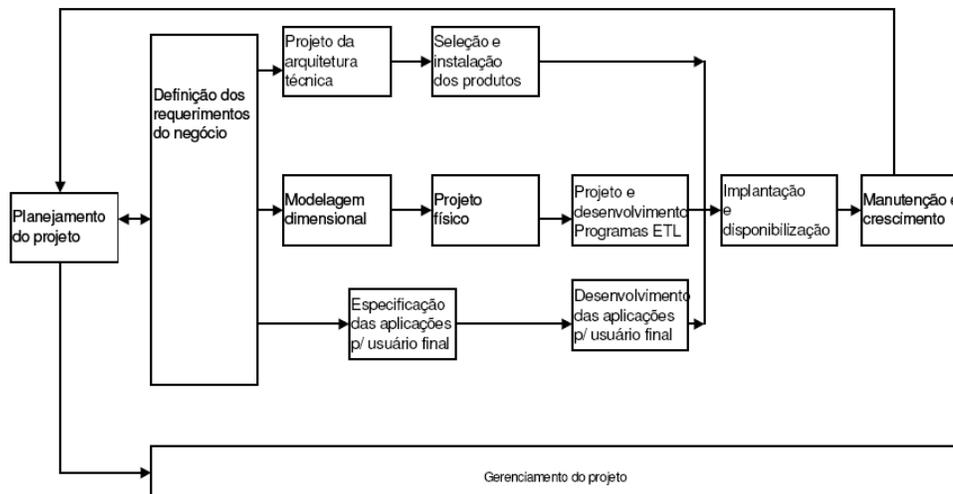


Figura 7 – O processo de *DWing*.

Fonte: Kimball *et al.* (1998)

Após a descoberta dos requisitos há três conjuntos de tarefas no ciclo de vida. O primeiro é responsável pelo desenho do modelo dimensional do DW. O segundo, responsável pela concepção tecnológica (ou seja, pela estrutura física do banco de dados), fica encarregado de definir e desenvolver os programas responsáveis pela extração, transformação e carga (do inglês ETL) do DW a partir dos sistemas transacionais ou fontes externas. Por fim, no terceiro e último conjunto de tarefas, contempla-se a perspectiva da aplicação, ou seja, a especificação dos relatórios, consultas e cálculos requeridos pelos usuários. O final do ciclo de vida é a manutenção e o crescimento do DW, dado que esse é um tipo de sistema que deve evoluir dinamicamente de acordo com o ambiente de negócio e suas necessidades de tomada de decisão. Assim, o ciclo de vida exige um novo ciclo de desenvolvimento e aperfeiçoamento, que tem como ponto de partida o planejamento do novo projeto (Penna, 2004).

O resultado do processo de *Data Warehousing*, na visão do autor Kimball *et al.* (1998), é um sistema DW que está em constante evolução e que tem como principal objetivo prover os decisores de informações para a tomada de decisão. Swift (2001), no entanto, afirma que a maior importância de um *data warehouse* está na sua capacidade de prover informações para subsidiar o reprojeto dos processos de negócio, ou seja, a melhoria desses. Esse tipo de sistema, na concepção do autor, permite a avaliação dos processos de negócio dado que disponibiliza as informações necessárias para tal ação.

Nesse ponto, faz-se necessário destacar que o processo de construção de DW, tal como demonstrado acima, se diferencia do processo tradicional de

desenvolvimento de sistemas transacionais. No entanto, em ambos os casos há a fase de definição dos requisitos para os sistemas. Na próxima seção, será feita uma breve revisão bibliográfica acerca do tema.

3.3 Requisitos para Data Warehouses

As atividades de levantamento e análise de requisitos geralmente estão presentes na etapa de definição de *softwares* transacionais e analíticos. Como essa é uma fase que apresenta uma série de dificuldades, pois há o reconhecimento de que é na atividade de descoberta que surgem os problemas mais dispendiosos e de maior impacto negativo (Alves²¹ *apud* Cruz, 2004), criou-se um campo específico dentro da Engenharia de *Software* (ES) para tratar dessa etapa e de suas questões. Segundo Azevedo Junior & Campos (2008), a Engenharia de Requisitos (ER) visa aplicar técnicas de engenharia em métodos de definição e análise de requisitos para garantir o atendimento das necessidades de informatização de processos através do *software* projetado. Isto é, dentro da ES é a Engenharia de Requisitos (ER) que lida com uma parte fundamental no processo de produção de *software* – a definição do que se pretende produzir. Logo, cabe a ER propor métodos, técnicas e ferramentas para facilitar o trabalho de definição daquilo que se espera de um sistema de informação.

Inúmeras definições são encontradas na literatura para a ER (Zave, 1997; IEEE, 1998; Leite, 1987; Hofman, 1993; Davis, 1993; Kotonya *et al.*, 1997; Lamsweerde, 2000). No entanto, a definição adotada nesse trabalho é a de Leite (1994), pois fornece um entendimento objetivo e completo do termo ao afirmar que a ER é a disciplina que procura sistematizar o processo de definição de requisitos. Ainda segundo o mesmo autor, a ER estabelece o processo de definição de requisitos, no qual deve fazer a elicitação, modelagem e análise dos requisitos. Esse processo deve trabalhar com diferentes pontos de vista e usar uma combinação de métodos, ferramentas e pessoal, pois é um processo que ocorre num contexto previamente definido e chamado Universo de Informações (UdI). Assim, a ER propõe um conjunto de métodos, técnicas e ferramentas para apoiar o processo de definição de requisitos.

²¹ Alves, C. F. *Seleção de Produtos de Software Utilizando uma Abordagem Baseada em Engenharia de Requisitos*. Dissertação (Mestrado em Informática). Centro de Informática - Universidade Federal de Pernambuco, Recife, 2001.

Pressman (2002) sugere, conforme ilustrado na Figura 8, as etapas da ER, sob o ponto de vista do conhecimento que está sendo trabalhado. Inicialmente, o universo de informações é analisado, conforme consta na definição de Leite (1994), para detecção das necessidades de negócio e identificação dos requisitos provenientes desse universo (muitas vezes intrínseco ao negócio). Na seqüência, ocorre a fase de especificação, na qual as informações coletadas são modeladas e especificadas através de representações (modelos ou diagramas). Ao final, as representações sofrem validações e, quando necessário, reinicia-se o processo para obter informações complementares. O produto final da ER é o documento de requisitos de *software*.

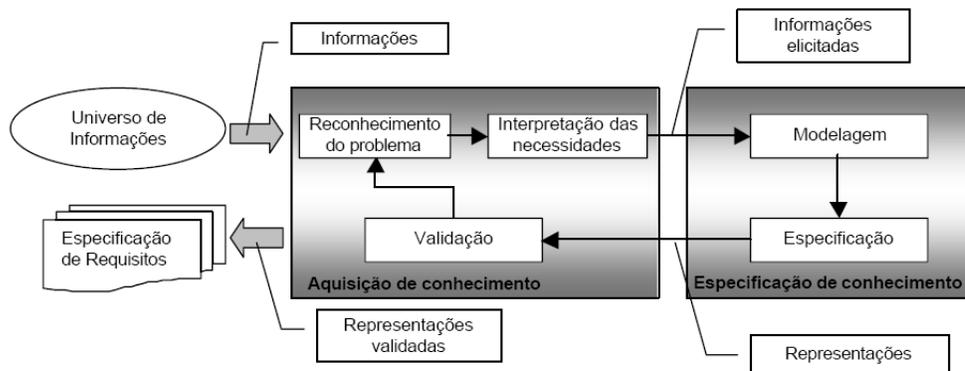


Figura 8 – Processo da Engenharia de Requisitos

Fonte: Pressman (2002)

Para Leite (1994) as fases da ER estão organizadas da seguinte forma:

- **Elicitação:** tarefa de identificação dos fatos que compõem os requisitos do sistema, visando prover um entendimento do sistema mais correto e completo;
- **Modelagem:** consiste na representação dos fatos extraídos da etapa anterior num modo sistemático e através de técnicas específicas;
- **Análise:** é o processo de verificação e validação do resultado da modelagem, ou seja, é a análise da especificação.

Existe ainda uma fase da ER chamada gerenciamento de requisitos que ocorre ao longo das demais etapas. Sommerville *et al.* (1993) definem o gerenciamento de requisitos como a etapa responsável pelo controle das mudanças de requisitos durante o desenvolvimento do sistema, caracterizando-se como um processo importante dado o caráter geralmente incompleto,

incorreto e inconsistente dos requisitos. A gerência de requisitos garante a rastreabilidade na evolução dos requisitos. Esse tema não será explorado por não estar contido no escopo da pesquisa, mas é importante mencionar sua existência no processo da ER.

Antes de aprofundar a explicação sobre os demais conceitos da ER, cabe salientar que a sistematização da etapa de definição de requisitos, atividade fundamental da ER, é necessária, porque a complexidade dos sistemas atuais exige maior atenção ao correto entendimento do problema a ser solucionado pela ferramenta de *software*. A etapa de definição de requisitos é, portanto, considerada como a atividade mais importante, decisiva e ao mesmo tempo crítica no desenvolvimento de sistemas, principalmente no que tange à elicitação dos requisitos (Christel & Kang, 1998). Esse será o próximo conceito a ser apresentado.

É comum ver profissionais da ES ou da ER se deparando freqüentemente com usuários e clientes insatisfeitos com o produto de *software* desenvolvido. Isso porque muitas vezes os projetos de desenvolvimento de sistemas ocorrem sem o claro entendimento e, conseqüente, especificação de requisitos coerentes e aderentes às reais necessidades dos clientes e de seus usuários finais (Maffeo, 1992). Por isso, a comunidade da ES cada vez mais volta sua atenção e esforços para o processo de elicitação de requisitos na tentativa de minimizar seu caráter frágil.

Para Sommerville & Sawyer (1999) existem quatro dimensões na elicitação de requisitos:

1. Entendimento do domínio da aplicação, ou seja, ter o conhecimento geral da área onde o sistema é aplicado;
2. Entendimento do problema, isto é, entender os detalhes do problema específico do cliente;
3. Entendimento do negócio para compreender como o sistema pode contribuir para o desenvolvimento do negócio. É importante identificar como o *software* afeta e interage com as diferentes partes da organização e como ele pode melhorar as expectativas do negócio;
4. Entendimento das necessidades e restrições dos interessados no sistema.

Na ER é tarefa da elicitação de requisitos realizar a identificação dos fatos que conformam os requisitos do sistema, gerando entendimento mais completo e

correto das características e funções do *software*. Nesse processo de descoberta de informações é preciso habilidade tanto para lidar com especialistas humanos quanto para extrair informações relevantes impregnadas no conhecimento tácito (Goguen & Jirotko, 1994).

Leite (1994) define a elicitação de requisitos como o conjunto de atividades de descoberta de requisitos dos sistemas, identificação das fontes de informação, coleta de fatos e comunicação. Rzepka (1989) decompõe a elicitação de requisitos em:

- Identificação das partes relevantes que são fontes potenciais para a extração de requisitos como usuários finais, sistemas de interface e fatores do ambiente circundante;
- Coleta de informações para compor a “lista de desejos” das partes relevantes. Essa primeira lista pode conter ambigüidades, instabilidades, inconsistências, bem como incompletude;
- Documentação e refinamento da “lista de desejos”, incluindo todas as atividades e dados considerados importantes. Nesse ponto, as necessidades ainda estão descritas em alto nível e sem rigoroso formalismo técnico;
- Integração da lista de necessidades dos diversos envolvidos, eliminando conflitos e inconsistências. O emprego de modelos para representar as informações coletadas na fase de elicitação pode auxiliar no entendimento das necessidades do sistema;
- Determinação dos requisitos não-funcionais²² como performance e confiabilidade.

Ainda segundo Rzepka (1989), a maioria dos processos de elicitação de requisitos encontrados na literatura contém as atividades supracitadas. Contudo, os meios para execução dessas tarefas e suas iterações ainda não estão bem

²² A complexidade de um *software* é proveniente tanto das suas funcionalidades (o que o sistema deve fazer) quanto de requisitos mais gerais como custo, performance, confiabilidade, manutenibilidade, portabilidade, custos operacionais entre outros. Esses são os requisitos não funcionais que abrangem inclusive aspectos de qualidade, restrições, objetivos, entre outros. Assim como os funcionais, os requisitos não funcionais desempenham um papel crítico durante o desenvolvimento de sistemas. E erros provenientes da não elicitação ou da elicitação incorreta são caros e difíceis de corrigir após a implementação do sistema (Davis, 1993).

entendidas e isso, de certa forma, se traduz em percalços no cumprimento do principal objetivo da primeira etapa da ER.

3.3.1 Definindo Requisitos

O termo “requisitos” tem sido bastante ao longo desse trabalho como sinônimo para necessidade do sistema, mas nessa seção serão apresentadas definições encontradas na literatura, a saber:

“Um requisito é algo que uma aplicação de computador deve fazer para os seus usuários. Essa é uma função ou característica ou princípio específico que o sistema deve prover. Os requisitos constituem o escopo do projeto de desenvolvimento do *software*.” (Kulak, 2001, p.4)

“(1) uma condição ou capacidade necessária para um usuário resolver determinado problema ou atingir um objetivo; (2) uma condição ou capacidade que um sistema deve ter ou prover... para satisfazer um contrato, padrão, especificação ou outro documento formal imposto.” (IEEE²³ *apud* Davis, 1982, p.15)

“Requisitos são definidos durante os primeiros estágios do desenvolvimento de um sistema como uma especificação daquilo que deve ser implementado. Eles são descrições de como o *software* deve se comportar ou das propriedades e atributos do sistema. Eles podem ser restrições do processo de desenvolvimento de um sistema...” (Sommerville & Sawyer, 1999, p.13)

Goguen (1994) define requisito como propriedades de um sistema responsáveis pelo êxito no ambiente em que será utilizado. Para outros autores como Macedo & Leite (1999) e Gilb (1999), os requisitos de *software* são tanto necessidades funcionais (comportamentos e propriedades do sistema), quanto necessidades não funcionais, ou seja, os quesitos de qualidade e restrições operacionais ou do processo de desenvolvimento do sistema. Já Leite (2001) define requisitos de *software* como sentenças que expressam as necessidades dos clientes e que determinam a qualidade do sistema.

A definição adotada nesse trabalho para requisitos é a Gottesdiener (2003) que os define como capacidades operacionais que um sistema ou processo deve

²³ Institute of Electrical and Electronics Engineering. **IEEE Standard Glossary of Software Engineering Terminology**. ANSI/IEEE Standard 729-1983. New York, 1983.

ter para satisfazer uma necessidade do negócio. Essa definição foi adotada por abordar objetivamente a influência de aspectos do negócio no sistema. Os requisitos, algumas vezes também chamados requerimentos, servem como base para o desenvolvimento de todo e qualquer sistema. O termo “requisito” abrange tanto os aspectos funcionais quanto os não funcionais dos *softwares*. Os requisitos constituem o produto final de um processo sistemático de descoberta e definição e devem, ao final desse processo, ser claros e precisos. Ainda de acordo com a mesma autora, os requisitos de *software* podem ser vistos sob três perspectivas: a do negócio, do usuário e a da técnica.

Os requisitos de *software* também recebem classificações para facilitar o seu entendimento. Segundo Leite (2001), os requisitos podem ser classificados em funcionais, que estão diretamente ligados à funcionalidade do sistema; e não funcionais, que expressam restrições que o sistema deve atender ou qualidades específicas que o mesmo deve ter. Sommerville (2003) afirma que os requisitos funcionais são declarações de funções fornecidas pelo sistema. Essas declarações dizem como deve ser a reação do sistema mediante entradas específicas e como deve ser seu comportamento em certas situações. Em algumas destas, requisitos desse tipo também podem expressar restrições explícitas daquilo que o sistema não deve fazer. São exemplos de requisitos funcionais: calcular gastos mensais, emitir relatórios de compras quinzenalmente, notificar clientes em situação de inadimplência, impedir compra de itens por clientes em débito.

Embora esse trabalho não tenha como foco os requisitos não funcionais, cabe mencioná-los para tornar mais completa a abordagem dessa seção de revisão da literatura. Segundo Sommerville (2003), esses requisitos constituem restrições sobre serviços ou funções fornecidos pelo sistema. Esses tipos de requisitos aparecem de acordo com a necessidade dos usuários seja devido a restrições de orçamento, de políticas organizacionais ou à necessidade de interoperabilidade com outros sistemas, bem como por necessidades do negócio. Os requisitos não funcionais são agrupados em três tipos principais: de produto, organizacionais e externos. Uma taxonomia de requisitos não funcionais é esquematizada na figura que se encontra no Anexo 1.

Os requisitos de produto especificam o comportamento do sistema (p. ex.: requisitos de desempenho – rapidez com que o sistema opera e a necessidade de memória; requisitos de confiabilidade – taxa aceitável para falhas). Já os requisitos organizacionais são oriundos de políticas e procedimentos organizacionais (p. ex.: requisitos de padrões – tipo de processo de

desenvolvimento a ser adotado; requisitos de implementação – linguagem de programação adotada; requisitos de entrega – prazos de entrega do produto e de seus documentos). Por fim, destacam-se os requisitos externos que abordam fatores externos ao sistema e ao seu processo de desenvolvimento (p. ex: requisitos de interoperabilidade – dizem como o sistema deve interagir com outras partes da organização; requisitos legais – características do sistema para atender às leis; requisitos éticos – garantem aceitabilidade do sistema para seus usuários) (Knight, 2004).

A diferença entre um requisito funcional e um não funcional nem sempre é clara. Essa confusão muitas vezes ocorre porque os requisitos não funcionais geralmente relacionam-se com um requisito funcional (Eagle²⁴, Chung *et al.*²⁵ *apud* Cysneiros, 1997). A partir da definição, de forma mais simplória, pode-se dizer que um requisito funcional representa algum tipo de transformação no sistema, já o requisito não funcional apresenta a forma como essa transformação se comportará e quais serão as suas qualidades específicas.

Além da classificação dos requisitos em funcionais e não funcionais, essas duas classes ainda podem se encontrar divididas em níveis. Vicente (2004) afirma que essa abordagem ajuda a gerenciar a complexidade dos mesmos, além de facilitar a troca de informações entre os diversos colaboradores do desenvolvimento (engenheiros de *software*, usuários e gerentes) e os *stakeholders*, proporcionando diferentes níveis de detalhamento. Gottesdiener (2002) faz a segregação dos requisitos em dois níveis: do usuário e dos requisitos do sistema. Os primeiros constituem declarações sobre as funções ou restrições do sistema em alto nível, devendo ser uma especificação consistente do comportamento externo do sistema. Por outro lado, os requisitos do sistema definem de maneira bem detalhada as funções e restrições sob a ótica do sistema, gerando uma especificação consistente e bem completa daquilo que o sistema deve executar.

Há ainda um nível superior a esses dois (funcionais e não funcionais), chamado nível dos requisitos de negócio (Sommerville, 2003). Esses requisitos são as descrições das necessidades que existem no negócio executado pelos usuários. Os requisitos de negócio abordam os objetivos, processos, papéis e

²⁴ EVALUATION OF NATURAL LANGUAGE PROCESSING SYSTEMS (EAGLE). 1995. Disponível em: <www.issco.unige.ch/ewg95>

²⁵ CHUNG, L., NIXON, B., YU, E., MYLOPOULOS, J. **Non-Functional Requirements in Software Engineering**, Kluwer Academic Publishers, 1999.

outras características da organização que está construindo o sistema. Esses requisitos apresentam, num nível mais alto, as funções do sistema descritas, cumprindo assim a finalidade do sistema – atender às necessidades do negócio.

Knight (2004) apresenta exemplos dos diferentes níveis de requisitos:

- Requisito de negócio: “Aumentar o controle sobre as vendas de uma loja”. Representa uma necessidade da organização e, que por uma decisão interna, será atendida por um sistema;
- Requisitos de usuário: “Cadastrar as vendas feitas em cada loja”, “Registrar devoluções” e “Buscar o total de vendas feitas em todas as lojas”. Descrevem funções executadas pelos usuários, visando o atendimento das necessidades da organização;
- Requisitos de sistema: “Registrar cada compra, armazenando o seu cliente, vendedor, data, itens comprados e a loja que vendeu”, “Registrar a devolução indicando a compra, motivo e data”, “Consultar o total de vendas de cada loja e fornecer um total”. Esses requisitos apresentam funções e restrições do sistema que atendem aos requisitos de usuário.

Essa divisão em níveis auxilia o gerenciamento de requisitos e sugere que o processo de elicitação seja interativo, iniciando-se pelos requisitos de negócio e desdobrando-se posteriormente em requisitos de usuário e de sistema, respectivamente. A Figura 9 esquematiza esses desdobramentos.



Figura 9 – Hierarquia de requisitos

Fonte: Adaptado de Knight (2004)

A própria hierarquia de níveis ressalta a importância do conhecimento do negócio, ou seja, o aprendizado sobre a organização e as suas tarefas para o desenvolvimento de sistemas. Vale destacar que nesses níveis os requisitos não estão descritos em detalhes técnicos, sendo importante apenas determinar aquilo que o sistema deve fazer e o porquê dessa ação, sem que haja maiores preocupações de como fazer (esse detalhamento só ocorre após a descoberta do requisito de sistema) (Knight, 2004).

No caso de sistemas analíticos, como *data warehouses*, os requisitos se relacionam com o comportamento funcional e a disponibilidade de informações para a tomada de decisão, englobando aspectos de acessibilidade, transformação, organização e agregação das mesmas. Assim, os requisitos de DW, quando identificados corretamente, habilitam os decisores a estabelecerem direções e expectativas relacionadas aos objetivos organizacionais, já que terão as informações necessárias para tal (Bruckner *et al.*, 2001).

Os requisitos de DW também podem se apresentar em diferentes níveis de abstração, seguindo a hierarquia representada na Figura 9. No caso de sistemas DW também há três níveis, a saber (Bruckner *et al.*, 2001):

- Requisitos de negócio: representam os objetivos organizacionais em alto nível. Esses requisitos são geralmente obtidos a partir dos elementos estratégicos da empresa (missão, visão e valores). Esse tipo de requisito captura os benefícios primários do DW para a organização, expressando oportunidades do negócio, usuários típicos do sistema DW e as necessidades organizacionais em alto nível de abstração. O trabalho proposto tem foco nesse tipo de requisito, manipulando-o de forma mais direta;
- Requisitos dos usuários: descrevem as tarefas que os usuários devem estar aptos a realizar a partir do auxílio fornecido pelo DW. Esses requisitos devem coletados diretamente com os decisores que usarão o DW, já que esses conhecem as principais tarefas decisórias a serem executadas, bem como aspectos não funcionais do sistema DW. Importante lembrar que os requisitos dos usuários devem estar sempre alinhados aos requisitos de negócio estabelecidos inicialmente;
- Requisitos de sistema: representam os requisitos finais do DW bem detalhados, ou seja, aqueles que servirão para a equipe de desenvolvimento como base para as demais tarefas do processo de construção do DW. Esses requisitos devem estar alinhados àqueles definidos anteriormente (de negócio e dos usuários). Requisitos

funcionais e de informação encontram-se entre os requisitos de sistema. Os primeiros expressam serviços e funções disponíveis no futuro sistema que permitirão aos usuários cumprirem suas tarefas (análises) no DW, satisfazendo os requisitos dos usuários. Já os segundos referem-se às necessidades de informação da organização, descrevendo as informações e dados que o DW deve fornecer, bem como aspectos relacionados à qualidade, acesso, processamento e combinação dos mesmos. Outros aspectos relacionados à interface e ambiente (cultura, política, regulamentações) também podem ser referenciados nesse nível da hierarquia dos requisitos.

A correta identificação dos requisitos em seus diversos níveis hierárquicos só é possível quando há um processo de elicitação de requisitos muito bem estruturado, pois, dessa forma, evita-se que ao final do projeto de desenvolvimento tenha-se um sistema DW que não atenda às necessidades dos tomadores de decisão. No entanto, a tarefa de elicitação não é trivial e nos últimos anos vem recebendo atenção e contribuições importantes, conforme será visto na próxima seção.

3.4

O Problema da Elicitação de Requisitos para *Data Warehouse*

O processo de construção de DW é difícil e, por inúmeros motivos, alguns projetos de DW acabam falhando. Estudos mostram que 40% dos projetos de DW não são finalizados e 85% falham no atendimento aos objetivos do negócio (Schneider *et al.*, 1999).

As falhas de projetos de DW podem ser atribuídas, por exemplo, à dificuldade de identificação das necessidades de informação dos potenciais usuários do DW, pois esses geralmente não sabem formular bem as suas necessidades. Além desse problema, há outros considerados clássicos na tarefa de elicitação de requisitos como a disparidade de linguajar entre usuários e equipe técnica, dificuldade de entendimento, conflitos de interesses, volatilidade dos requisitos, complexidade, ambigüidade dos requisitos, dentre outros (Boehnlein & Ulbrich Vom Ende, 2000). Muitos desses problemas ocorrem na fase inicial do processo da ER, ou seja, na fase de elicitação (definição) dos requisitos. Dessa forma, há um consenso entre diversos autores de que o insucesso dos projetos de DW se deve a erros na fase de definição dos

requisitos (Cabibbo & Torlone, 1998; Lehner *et al.*, 1998; Tryfona *et al.*, 1999; Vassiliadis, 2000).

Além dos erros típicos do processo da Engenharia de Requisitos, pode-se destacar a diferença de objetivos almejados nessa fase inicial entre sistemas analíticos e transacionais convencionais. Nos projetos de construção de DW, o objetivo principal é a descoberta de dados e fontes de informação, envolvendo objetivos e estratégia organizacional (Boehnlein e Ulbrich Vom Ende, 2000). Já no caso de projetos de desenvolvimento de sistemas transacionais o foco é a descoberta de funcionalidades do sistema (Prakash & Gosain, 2008).

Outra questão a ser observada em projetos de construção de DW é o fato de projetos desse tipo muitas vezes não realizarem a tarefa de elicitação de requisitos, passando diretamente para a fase de *design* e modelagem de dados. Assim, a falta de uma visão de alto nível dos requisitos do DW muitas vezes acaba comprometendo o sucesso do projeto, dado que aspectos como necessidades dos usuários, restrições multidimensionais e de qualidade podem não ser consideradas.

Segundo Paim (2003), apesar da existência de abordagens que abrangem aspectos do ciclo de desenvolvimento de DW, a maioria apresenta oportunidades de melhoria em relação ao início, que é a fase de levantamento (definição ou elicitação) de requisitos. Essas melhorias estão relacionadas principalmente à possibilidade de sistematização do método aplicado à análise dos requisitos de negócio que irão nortear a construção da aplicação.

Nesse sentido, ter diretrizes que forneça uma plataforma em alto nível para suporte ao projeto de sistemas DW é uma necessidade premente (Tryfona *et al.*, 1999). Na concepção do autor ter uma visão de alto nível dos requisitos do DW é a chave para tornar os projetos de *data warehouse* mais próximos do entendimento do usuário e garantir a independência de aspectos de implementação.

Boehnlein & Ulbrich Vom Ende (2000) confirmam a relação entre sistemas DW e o negócio ao afirmarem que o DW é projetado de acordo com um conjunto relevante de assuntos afetos aos processos de negócio. Assim, um DW é concebido e construído para dar respostas aos decisores acerca das questões afetas aos processos de negócio, saindo do foco das transações individuais (Adamson & Venerable, 1998). Por sua vez, os mesmos autores afirmam ainda que o processo de tomada de decisão necessita de métricas, ou seja, de indicadores para monitorar e avaliar se os processos de negócio estão contribuindo para o alcance dos objetivos da empresa. Esse monitoramento se

torna possível através do DW, que passa a ser importante instrumento tecnológico para viabilizar a geração de informações afetas a métricas de desempenho. Essas métricas possibilitam verificar se os processos de negócio estão alinhados aos objetivos estratégicos, mostrando a forte relação entre objetivos organizacionais, indicadores de desempenho, processos de negócio e requisitos do DW.

É para aprimorar essa relação que o presente trabalho busca tecer suas considerações. A proposta desse trabalho se baseia na possibilidade de aplicar algumas diretrizes (heurísticas) capazes de desdobrar os indicadores de desempenho desde os objetivos estratégicos até o nível dos processos de negócio, visando construir os principais requisitos do DW ainda em alto nível. O foco nesse caso é a perspectiva do negócio, ou seja, não são levados em consideração apenas os requisitos tecnológicos e dos usuários, mas principalmente os indicadores de desempenho e seus processos de negócio, que, na prática, contribuem para a identificação das fontes de informação do DW.

3.5 Considerações Finais

Nesse capítulo foram apresentados os principais conceitos afetos à Engenharia de Requisitos que, por sua vez, contém a tarefa de elicitação de requisitos – uma das mais críticas atividades do processo de desenvolvimento de *software* e foco do presente trabalho. Também foram abordados conceitos sobre sistemas *data warehouses* e seu processo de construção – *Data Warehousing* (DWing) – com o objetivo de esclarecer que o processo de desenvolvimento de sistemas DW engloba muitas atividades, sendo a elicitação de requisitos, a primeira de todas e uma das mais difíceis por inúmeras razões. Por fim, esse capítulo abordou conceitos sobre requisitos de sistema, seus tipos (funcionais e não funcionais), seus níveis de abstração (requisitos de negócio, de usuário e de sistema), bem como a aplicação dessa hierarquia ao caso de sistemas DW.

A revisão bibliográfica afeta aos temas supracitados torna-se importante para facilitar o entendimento do leitor sobre a “interseção” da EPN e da Gestão de Desempenho no contexto da ER aplicada ao processo de construção de DW, relacionando-se, assim, ao objeto de pesquisa (definição de requisitos de DW a partir de indicadores de desempenho).